

Software Reuse FAQ

Frequently asked questions about software reuse.

What is software reuse?

Software reuse is the reapplication of a variety of kinds of knowledge about one system to another system in order to reduce the effort of developing and maintaining that system. Software reuse is typically achieved by the reapplication of artifacts produced during the development of one system to the development of another system.

What is the difference between an artifact, an asset, and a component?

An artifact is any item produced at some point during the software development life cycle. An asset is an artifact that is recognized as having a particular value. A software component is a clearly delineated piece of software that performs a useful function within a software system.

What is a reusable software component (RSC)?

A reusable software component, or RSC, is a software entity intended for reuse. It may be the design, the code, or some other product of the software development process. RSCs are sometimes called software assets.

What artifacts may be reused?

In principle, many different artifacts produced during the software development life cycle can be reused. Some typical examples of reusable artifacts include: source code, analysis and design specification, plans (project management), data (testing), documentation, expertise and experience (life cycle model reuse, quality assurance), and any information used to create software and software documentation. However, while all of these items are useful, the most often reused artifacts are software components.

What are some typical characteristics of a reusable software component?

Typical characteristics of a reusable software component are:

- It is self-contained with clearly defined boundaries.
- It is easily integrated with other components.
- It has a clear, unambiguous, and understandable specification.
- It conforms to its specification.
- It is verifiable -- the user should be able to test that the component conforms to its specification.
- The implementation details are hidden through encapsulation.
- It has a clearly defined interface.

What items are useful to package with reusable components?

Useful items to package with reusable components to help improve their reusability are:

- Documentation
- A test or example application
- Installation guides or scripts

What are the major steps in a software reuse process?

Some important steps are:

- Specifying the object to be created
- Searching the project, domain, and general databases for reuse candidates
- Evaluating the candidates to determine which (if any) should be used
- Modifying, if necessary, to fit specific needs
- Integrating the reusable component(s)

- Validating the system including the new component(s)
- Feeding back the knowledge regarding the payoff of reuse

What is the difference between black-box and white-box reuse?

Black-box means that the code implementation is hidden from the end user. Black-box components are typically distributed as binaries. Well-defined and documented interfaces allow the consumer to use the component without needing to know how it has been implemented. Black-box components can be more expensive to produce than white box components, but they also deliver greater overall savings.

White-box reuse is reuse with modification where the user has access to the internal implementation of the component. It is more popular with many implementers because the component can be tailored to fit the exact needs of the target system. However, the productivity benefits of reuse can rapidly diminish as more modifications are introduced. In addition, changes to the code can introduce errors or other unexpected side effects.

What is reusability?

Reusability is the extent to which a software component is able to be reused. Conformance to appropriate design and coding standards increases a component's reusability.

What is portability?

Portability is the extent to which a software component originally developed on one computer and operating system can be used on another computer and/or operating system. A component's reusability potential is greater if it is easily portable.

What is a domain?

A domain is a class of related software applications. Domains are sometimes described as "vertical", addressing all levels of a single application area (e.g., command and control), and "horizontal", addressing a particular kind of software processing (e.g., data structure manipulation) across applications. The potential for reuse is generally greater within a single domain.

What is domain analysis?

Domain analysis is the analysis of a selected domain to identify common structures and functions, with the objective of increasing reuse potential.

What is a library?

A library is a collection of reusable software components together with the procedures and support functions required to provide the components to users.

What is a retrieval system?

A retrieval system is an automated tool that supports the classification and retrieval of reusable software components. It is also called a repository.

What is a software lifecycle?

A software lifecycle is the series of stages a software system goes through during its development and deployment. While the specific stages differ from one project to the next, they generally include the activities of requirement specification, design, code, testing, and maintenance.